

W - A - R

"New driver... at last we meet!"

Declared: April 1986
Victory: September 1986

"An original version of the game was programmed and called 'Ad Infinitum', playing and looking like the game 'Uridium' with similar fonts and double scrolling.

The game was scrapped due to the similarities and this version was released with just single scrolling to cover up that it was a 'Uridium' clone."

Mobygames

Welcome to Rob Hubbard 3.0!

"W.A.R" is where Rob took his experience since mid-1985 and hugely upgraded the sophistication of his driver, expanding the cleverness of routines, and streamlining the code to make it more flexible and remove hardcoded effects.

Some effects were removed as he'd grown out of them.

Read on!

"This was a game by Stoa and Tim, who were prominent C64 sceners.

The whole C64 music and games culture was changing round about this time, and we all started doing more adventurous things.

This tune was something I had also written a few years prior, but had simply shelved it. It is based on the concept of parallel moving triads, initially root position and later 2nd inversion. Much of this was the influence of Bartok and the Brecker Bros.

The tune gets into a jazz funk section. I used an idea that I had heard on a Corea album, where he 'trades fours' then twos, and then ones, and used a McCoy Tyner type 4ths lick to get out of that section.

I did an orchestral version for Back in Time with a guy called Steve Scherer back in the '90s and really didn't have much idea of how to set about handling the more complex parts, in terms of orchestration.

I revisited doing a complete new orchestration all these years later for 8-Bit Symphony, and it was so much easier and clearer what needed to be done to orchestrate the more harmonically complex sections.

Music is one of those subjects that you can never master – you can only progress."

Rob Hubbard

No self-respecting mention of "W.A.R" would miss out that entering "GO 159256" as your name in the high score entry would play a metallic version of the "Crazy Comets" high score.

In fact, the difference between this version and the original highlights how much the driver (and Rob's musical sensibilities) had changed since 1985.

This being the time of Compunet and parties, the "W.A.R" music hit Compunet almost as soon as it was finished (well, the title tune anyway as "W.A.R Demo").

Checking out the driver code between the Compunet demo and the final SID, there were no meaningful changes except the new SFX driver was deployed.

PORTS AND VERSIONS

No other ports had Rob's music in: given how mature it was, I'm not surprised they didn't try.

In fact, the game we know as "W.A.R" on the ZX Spectrum, CPC and BBC Micro was a different game altogether, with a tiny play area and vertical scrolling. It was like "Warhawk" in a window.

The BBC and CPC versions have some nice shooty effects, but it's the C64 version that stands out.

CRITICAL RECEPTION

It's fair to say that no one was particularly impressed by a delayed "Uridium" clone, no matter how famous the programmers were on Compunet. But then, given it had actually had a scroll direction disabled to prevent legal action and wasn't the game the programmers actually intended... maybe the lukewarm reviews weren't so surprising.

However, Rob's new, mature sound was just what the **ZZAP! 64** crew were looking for:

"JR: ... fabulous music...

Sound: 98%. Outstanding Rob Hubbard soundtrack and spot FX

Game: 44%"

C+VG seemed to think it was an Andrew Braybrook game, and the sequel to "Uridium". Weird.

*"The gameplay is astoundingly similar to 'Uridium'. The music is marginally superior to its predecessor. The sound effects and the high-score routine with scrolling letters are exactly the same. **Sound:** 7"*

7 for the music? Oh, C+VG....

W H A R W

(C) 1986 MARTECH

PRESS FIRE TO START

As I said, this is where Rob Hubbard 3.0 starts due to his substantially improved music and SFX drivers (which now decoupled the SFX driver from the music driver).

"Fine tempo" is back and large as life.

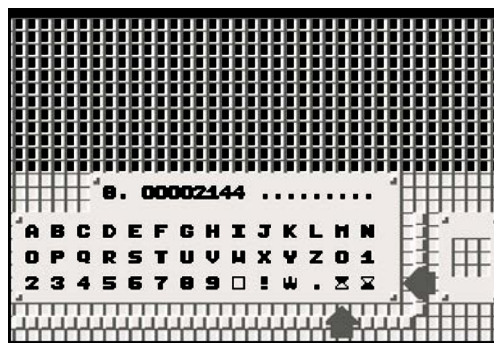
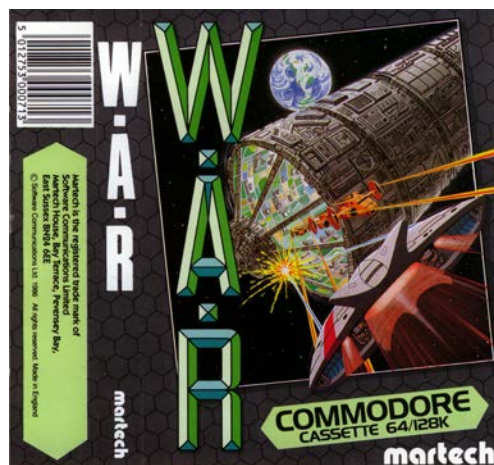
A fundamental change is that 16-byte patches have been split into two sets of 8-byte tables

The patch (mostly) holds the initial SID registers and an FX byte.

This contains parameters for the effects that might be applied to the sound.

Vibrato has changed again to one which has three main variables: vibrato count (keeps track of how long since it started), vibrato delay (how long the vibrato should wait before doing anything), and vibrato speed.

Pitch-bend has been improved and now self-modifies for efficiency.



FX DIRECTORY

The FX byte on the end of a patch specifies a combination of note effects. Each bit of the byte specifies a different effect, and each one is referred to in the code by its value rather than its position in the byte. Rob would add, change or remove these at will as the job demanded.

In fact, his driver is so flexible that about the only major changes he made before he added samples were adding new drums (for “Ace 2”, but these were really shown off in “Shockway Rider”), and filters. Some tunes such as “Delta” had extra features, but those features were game-specific.

Bit 1 is the drum effect and applies noise and a downward note slide (subtracting one from the frequency high byte each time).

Bit 2 is a wave interleave: every other frame it alternates the original waveform with an alternative one (but it doesn't try to combine them together).

Bit 4 replaces the first few frames of a patch with another waveform. This time it has variable delay and arbitrary waveforms. This is now how you can place noise at the beginning of a bass sound to create the illusion of simultaneous hi-hat hits.

Bit 8 alters the pitch at every frame, alternating from the current note to a separate fixed note specified in the sub-patch. There's no equivalent to this in the old driver.

Bit 16 is the chord handler, and it's markedly different from the one in “Hollywood or Bust”.

The code now has a small table of chords: each chord a collection of two “offsets” to apply to the actual played note.

For instance, “W.A.R” has the chord table of: \$FD, \$F9, \$FC, \$F7, \$0C, \$F4, \$00, \$00

The first chord is \$FD, \$F9. This means it's going to play notes below the original note (i.e. harmonies): values from \$00 to \$7F would be notes above the original. \$FD is three semitones below the original note, and \$F9 is 7 semitones below. If you played a C5 note, the others would be A4 and F4: i.e. it plays an F major chord.

“\$FC, \$F7” is -4 semitones and -9 semitones respectively. If you played a C5, the notes below it would be Ab4 and Eb4, so it's Ab Major chord but with what's known as a chord “inversion”, where the lowest note is not part of the chord name. A4, F5 and C6 is a “first inversion” of F major.

“\$0C, \$F4” is describing an octave above the original note, and an octave below. This is basically a super-duper double-height octave wobble. \0/

Bits 32, 64 and 128 are as yet unused... but watch this space!

Oh, and that “Crazy Comets” high score tune? The data was hidden away in fragments.

Voice 1 was hidden at the top of the data section, voice 2 was just above the playflag byte, and voice 3 was sandwiched between the patches and sub-patches (more to come on these).



But what is it good for? Answers on a postcard...